

# The Escalator Smart Switch

Submitted as part of the requirements for:  
CE713 Internet of Things

**Name:** Lorenzo Paris

**Lecturer:** Doc. Doctor Faiyaz

**Date:** 13 January 2018

## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	Brief Description.....	1
1.2	Tools Available.....	1
<b>2</b>	<b>THE SMART SWITCH.....</b>	<b>2</b>
2.1	The Algorithm.....	2
2.2	Important Considerations .....	2
2.3	How to Sense a Person ? .....	3
2.4	The Energy Saving Monitor .....	4
2.5	IoT Basic Elements .....	4
<b>3</b>	<b>IMPLEMENTATION .....</b>	<b>5</b>
3.1	Security Considerations .....	5
3.2	Software Source Code.....	6
3.3	Dedicated Web Page .....	8
<b>4</b>	<b>REFERENCES .....</b>	<b>8</b>

# 1 Introduction

## 1.1 Brief Description

The IoT Application I am going to describe is related to the concept of Smart City. Particularly, I would like to focus on a simple aspect which, from my personal point of view, is still too much underestimated.

My attention fell on big buildings endowed with escalators such as Shopping Centres which allow people to reach a floor. Until now we are used to see escalators working continuously, even without people standing on them. Therefore, the aim of my application is to make use of a sonar sensor in order to sense whether an escalator really needs to be activated. By doing this, a lot of energy can be saved and the data be displayed on a dedicated web page.

In order to make the application useful, interesting and commercial oriented, it will have to be as much self-standing as possible, meaning that it will work independently from ordinary variables such as engine, length or speed of the escalator itself.

The report will highlight, thus, all the steps required to implement what I called “*The Escalator Smart Switch*”.

## 1.2 Tools Available

Regarding the implementation of the aforementioned IoT application, I will make use of a LPC1768 board equipped with:

- mBED OS;
- Sonar Sensor;
- Network Board;
- 8 LEDs;
- 8 Switches;
- LCD Screen;
- Web Page (linked to the fixed IP Address of the Network Board).

# 2 The Smart Switch

## 2.1 The Algorithm

The algorithm of the Smart Switch is quite simple and can be explained in the following four steps:

1. Keep the escalator OFF;
2. Sense a person at the beginning of the escalator;
3. Turn ON the escalator until any person sensed by the sonar reaches the top (or the bottom) of the escalator;
4. Turn OFF the escalator.

The board will also be able to collect and compute the data provided by the sonar, such as the energy saved by this intelligent feature, and being transmitted to a Server. For a complete save energy management, the sensor itself could be fed by a rechargeable battery. A generator, therefore, can be linked to the timing belt of the escalator in order to produce that small amount of green energy which allows the whole board to work properly.

## 2.2 Important Considerations

The idea to activate the escalator when a person occurs needs to be extended and optimised.

To begin with, the board which includes the sonar sensor can be connected to the escalator power button in order to work continuously as a switch (ON/OFF).

The problem is, if the escalator starts working as soon as a person rises the first step, it may be dangerous as it can make lose the person's stability. Therefore, two possible solutions can be taken in consideration:

- The escalator starts working gradually:
  - **Pros:** better feeling for the person;
  - **Cons:** may be a more expansive solution; it is strictly dependent from this idea of intelligent solution (it requires the escalator's engine to be altered);
- The sonar sensor can be installed in a strategic position (**Figure 1**) to make the escalator starts working a few seconds in advance;
  - **Pros:** from the person's point of view it seems a usual escalator; furthermore, the sensor can be installed without making many changes to the escalator itself.

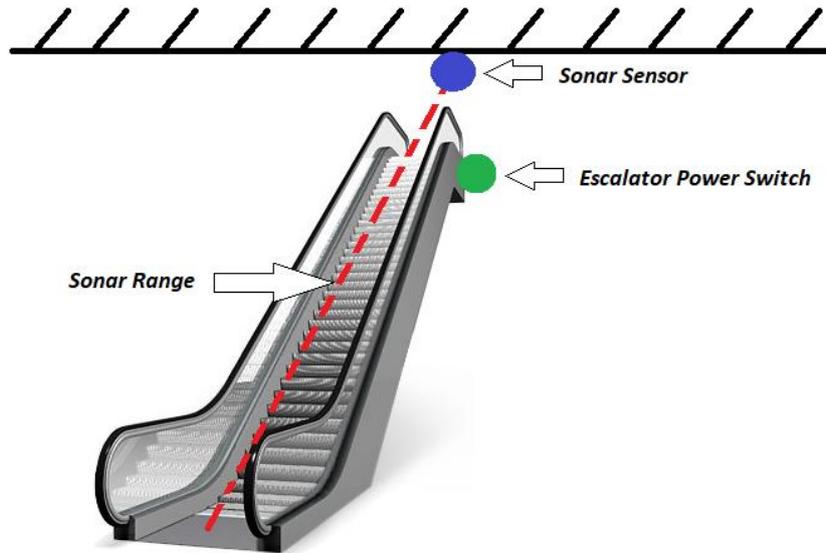


Figure 1 – Where to locate the Sonar

### 2.3 How to Sense a Person ?

After the installation of the board, in order sense a movement by the sonar, the idea is to take a first reading ( $D$ ) of the distance between the top and the bottom of the escalator. If  $D$  remains the same for a period of 10 seconds, it will be the reference distance (meaning the escalator is empty). From now on, every reading ( $D1$ ) which is less than  $D$  will be sensed as a person who needs to take the escalator. The latter, finally, will be turned OFF when  $D$  occurs again.

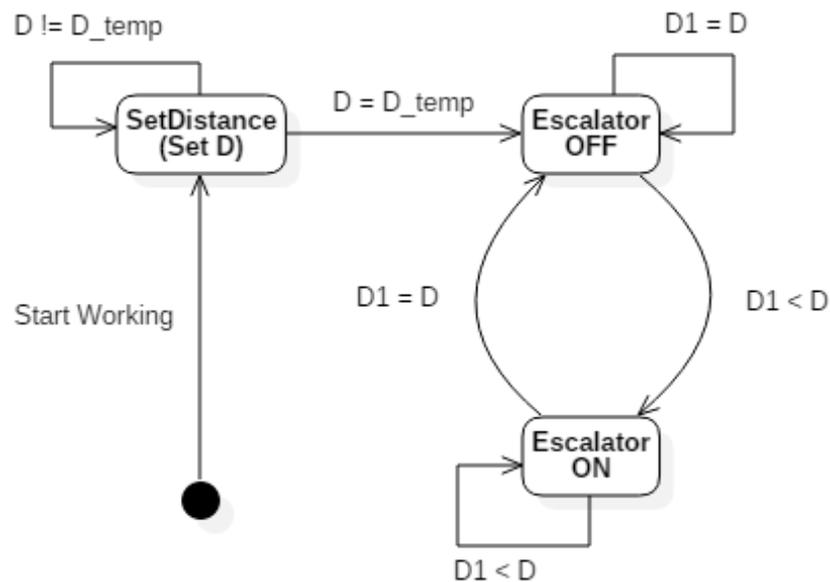


Figure 2 – State Machine

## 2.4 The Energy Saving Monitor

Any time a person is sensed by the sonar, the escalator will be turned ON and a stopwatch be started in order to count the time (**T\_ON**) the escalator has been made work. By comparing **T\_ON** with the total amount of time since all the system started working (**T\_tot**), it will be possible to compute the percentage of energy saved by the intelligent system. Previously, in fact, the escalator worked continuously without any break.

The board where the sonar is embedded into, include a Network board by which can be connected by an Ethernet Cable to the Internet. By doing this, all the data collected by the program, such as the time counter or the amount of energy saved, can be displayed remotely on a dedicated Web Page. The latter will allow on the main page to:

- Pick a particular escalator to view its statistics;
- Divide the escalator by sections/departments;
- Make data comparisons among them.

## 2.5 IoT Basic Elements



Figure 3 – IoT Basic Elements [1]

### IDENTIFICATION

- NAMING: Mac Address
- ADDRESSING: IPv4 (static)

### SENSING

- Sonar sensor

### COMMUNICATION

- By wire between the Sonar and the Switch
- Ethernet for the connection between the board and the Internet

### COMPUTATION

- HARDWARE: LPC1768 board
- SOFTWARE: mBED OS

### SERVICE

- Information Aggregation

### SEMANTICS

- Machine Learning

Every floor connected by escalators must have an even number of escalators (one towards each floor). This means we need an identification system which allows to discretise remotely which board (that includes the sonar) we want to analyze. At this scope, the MAC Address of each Network board can be used to differentiate a sonar sensor to another within the building. In order to view on Internet the data collected by each sonar, we can take advantage of the IPv4 addressing method assigned to each board. By visiting a particular IP address, a user will be able to log into a web page related to that board. Regarding the connection between the sonar and the switch, they can be easily connected by a simple wire (dual signal: ON/OFF) or wirelessly through a Bluetooth connection (more expensive solution). The data collected can be centralised and processed by algorithms related to Information Aggregation; then it can be processed through Machine Learning algorithms in order to figure out patterns and predict future behaviours as well as energy consumptions.

By using the aforementioned features related to the IoT application, we will have an IoT Reference Model given by a Physical Device & Controllers made up of the entire board which include the sonar sensor and the Network board. A Fog Layer can be depicted through the connection between the board and the Internet, which allows the view of the data collected by the Devices and processed through the computational level of the Cloud Layer.

## 3 Implementation

### 3.1 Security Considerations

The connections within the IoT application include the followings:

- By wire between the sonar sensor and the board;
- By wire between the board and the switch on the escalator;
- Ethernet cable between the data collected by the board and the Internet.

The only context which has the need to be protected seems to be the data sent to the Internet and displayed on the Web Page. A Man-In-The-Middle, in fact, might alter the data collected by the board and make the Web Page display different values.

By taking a cue from the mBED OS developer page, it provides a TLS library which includes a list of APIs for “protection against violations and mitigation of their consequences” [2]. Particularly, the Cryptographic library provides symmetric encryption and decryption functions by taking advance of the most commonly used algorithms, such as AES, which can be perfectly suited for the current application.

Unfortunately, due to several connection problems of the Network board, it was not possible to try and test such algorithms.

## 3.2 Software Source Code

The following section includes only part of the code used to implement *The Smart Escalator Switch* with the board provided. In particular, it is shown the main functions used to make the whole board work properly. However, due to connection problems of the Network board, it was not possible to try and test the section related to the data monitoring on the dedicated Web Page. Nevertheless, the same data can be displayed on the LCD screen embedded on the board by using the switches.

The whole project can be imported and tested on mBED Compiler from the following URL: <https://os.mbed.com/users/Elleppi/code/escalatorsmartswitch/>

```
#include "mbed.h"
#include "TextLCD.h"
#include "EthernetNetIf.h"
#include "HTTPServer.h"

/*Set reference distance: if the distance sensed by the sonar is the same for 9 seconds, it will be the reference distance, otherwise restart the cycle*/
int setDistance() {
    int D_temp = ReadSonar();
    int D;

    lcd.cls();
    lcd.printf("setting distance");

    SetLEDs(0);

    for(int i=0; i<9; i++) {
        D = ReadSonar();

        /*Allow an extra range of 10 mm for setting the distance*/
        if(D_temp - 10 < D && D_temp + 10 > D) {
            D_temp = D; //reset D_temp
            i = 0; //restart setting the distance
            SetLEDs(0);
        }
        else {
            wait(1);
            SetLED(i%9, 1);
        }
    }

    lcd.cls();
    lcd.printf("distance set at\n%d", D);
    wait(2);

    return D;
}

int main() {
    InitLEDs();
    InitWebServer();
    int D = setDistance();
    int D1;
    int T_ON = 0; //Time (seconds) the escalator is ON
    int T_tot = 0; //Time (seconds) since the system started working
    int timeSaved; //Time saved (seconds)
    int energySaved = 0; //Percentage of energy saved
    char ON = 0;

    char Buffer[4000]; //Used to write the HTML code on the Web Page

    while(1) {
        D1 = ReadSonar();

        /*ESCALATOR_POWER_ON*/
        //Allow an extra range of 10 mm when sensing the distance
        if(D1 < D - 10) {
            T_ON++;
            ON = 1;
        }
    }
}
```

```

/*ESCALATOR_POWER_OFF*/
else
    ON = 0;

T_tot++;
energySaved = 100 - ((100*T_ON)/T_tot);
timeSaved = T_tot - T_ON;

int firstOnSwitch = 0; //used to identify the switch

for (int a = 1; a < 9; a++ ) {
    SetLED (a, (ReadSwitch(a) + 1));

    if(ReadSwitch(a) == 1) {
        firstOnSwitch = a;
    }
}

/*Display the data collected on the LCD of the board*/
switch(firstOnSwitch)
{
    /*Switch 1: percentage of energy saved*/
    case 1:
        lcd.cls();
        lcd.printf("Energy Saved =\n%d%c", energySaved, '%');
        break;

    /*Switch 2: total elapsed time since the system started work-
ing*/
    case 2:
        lcd.cls();
        lcd.printf("T_tot:\n%d", T_tot);
        break;

    /*Switch 3: amount of time the escalator has been ON*/
    case 3:
        lcd.cls();
        lcd.printf("T_ON =\n%d", T_ON);
        break;

    /*Switch 4: amount of time saved (time the escalator has been
OFF) */
    case 4:
        lcd.cls();
        lcd.printf("Time saved =\n%d", timeSaved);
        break;

    /*No Switch: current state of the escalator (ON/OFF)*/
    default:
        if(ON == 1) {
            lcd.cls();
            lcd.printf("Escalator\nON");
        }
        else {
            lcd.cls();
            lcd.printf("Escalator\nOFF");
        }
}

/*HTML and CSS code to be sent to the dedicated Web Page*/
sprintf(Buffer, "<!DOCTYPE html><html><head><title>Escalator Smart
Switch</title></head><body><style type=\"text/css\">.tg {border-
collapse:collapse;border-spacing:0;border-color:#aabcfe;margin:0px auto;
}.tg td{font-family:Arial, sans-serif;font-size:14px;padding:10px 5px;border-
style:solid;border-width:1px;overflow:hidden;word-break:normal;border-
color:#aabcfe;color:#669;background-color:#e8edff;}.tg th{font-family:Arial, sans-
serif;font-size:14px;font-weight:normal;padding:10px 5px;border-style:solid;border-
width:1px;overflow:hidden;word-break:normal;border-
color:#aabcfe;color:#039;background-color:#b9c9fe;}.tg .tg-wfog{font-weight:bold;font-
size:22px;color:#f56b00;text-align:center;vertical-align:top}.tg .tg-3v14{background-
color:#D2E4FC;font-weight:bold;font-size:18px;text-align:center;vertical-align:top}.tg
.tg-5frq{font-style:italic;text-align:center;vertical-align:top}.tg .tg-
xqmu{background-color:#D2E4FC;font-style:italic;text-align:center;vertical-
align:top}</style>");

```

```

        sprintf(Buffer + strlen(Buffer), "<table class=\"tg\"><tr><th class=\"tg-wfog\"
colspan=\"2\">Escalator MAC Address</th></tr><tr><td class=\"tg-
3v14\">Description</td><td class=\"tg-3v14\">Value</td></tr>");
        sprintf(Buffer + strlen(Buffer), "<tr><td class=\"tg-5frq\">Working
Time</td><td class=\"tg-5frq\">%d seconds</td></tr>", T_ON);
        sprintf(Buffer + strl<tr><td class="tg-xqmu">Working Time Saved</td><td
class="tg-xqmu">%d seconds</td></tr>", timeSaved);
        sprintf(Buffer + strlen(Buffer), "<tr><td class=\"tg-5frq\">Energy
Saved</td><td class=\"tg-5frq\">%d%c</td></tr></table>", energySaved, '%');
        HTTPText(Buffer);
        HTTPText(Buffer);

        wait(1);          //Wait 1 second, then start again the loop
    }
}

```

### 3.3 Dedicated Web Page

Once the Network Board provided is fixed, the program will send the data to a dedicated Web Page by using the static IP Address of the Network board. The script includes a piece of HTML and CSS code which allows to display the data collected into a table. Here is a screenshot that represents how the Web Page would have been.

Escalator MAC Address	
Description	Value
<i>Working Time</i>	<i>15 seconds</i>
<i>Working Time Saved</i>	<i>156 seconds</i>
<i>Energy Saved</i>	<i>70%</i>

**Figure 4** – Dedicated Web Page Data Displaying

## 4 References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015. doi: 10.1109/COMST.2015.2444095.

[2] Arm MBED, "Mbed TLS". Internet: <https://www.mbed.com/en/technologies/security/mbed-tls/>.